PyPalEx

Generated by Doxygen 1.9.3

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Namespace Documentation

3.1 pypalex.__main__ Namespace Reference

Main script for PyPalEx.

Functions

- def thread_helper (extractor)

  Supports multiprocess color extraction operations.
- def **extract_color_palettes** ()

  Handles multiprocess color extraction from image(s).
- def set_global_args (args)

  Sets the global variables using the arguments.
- def check_source (image_path)

  Checks to make sure the path is either a file or directory.
- def check_sources (image_paths, directory=None)

  Checks each of the sources provided.
- def check_path (full_path)

  Check the full path to make sure the directory exists.
- def setup_argument_parser ()

  Sets up the argument parser for command line arguments.
- def **handle_args** ()

  Handles the arguments passed to PyPalEx.
- def **main** ()

  Main script function.

Variables

- list **EXTRACTORS** = [ ]
- list **PROPER_IMAGES** = [ ]
- list **OUTPUT_DIRS** = [ ]
- string **OUTPUT_DIR** = ''
- string **OUTPUT_TAIL** = "-color_palette.json"
- bool **PASTEL** = False
- bool **PASTEL_L** = False
- bool **PASTEL_N** = False
- bool **PASTEL_D** = False

3.1.1 Detailed Description

Main script for PyPalEx.

Author

Al Timofeyev

Date

February 2, 2022

This tool is for extracting color palettes from images.

3.1.2 Function Documentation

3.1.2.1 check_path()

```
def pypalex.__main__.check_path (
              full_path )
```

Check the full path to make sure the directory exists.

Parameters

full_path	The full path to directory.

Returns

Boolean flag: True if directory exists and is not a file, False otherwise.

3.1.2.2 check_source()

```
def pypalex.__main__.check_source (
              image_path )
```

Checks to make sure the path is either a file or directory.

Parameters

image_path	Path to image with filename and file extension (.jpg, .png, etc.).

Returns

Boolean flag: True if file exists, False otherwise.

3.1.2.3 check_sources()

```
def pypalex.__main__.check_sources (
              image_paths,
              directory = None )
```

Checks each of the sources provided.

Parameters

image_paths	Array of image paths.
directory	A directory path to the images, if it is provided.

Returns

Boolean flag: True if all/some sources are good, False if all sources are bad.

3.1.2.4 set_global_args()

```
def pypalex.__main__.set_global_args (
              args )
```

Sets the global variables using the arguments.

Parameters

args	User-supplied arguments.

3.1.2.5 setup_argument_parser()

```
def pypalex.__main__.setup_argument_parser ( )
```

Sets up the argument parser for command line arguments.

Returns

A command line argument parsing object.

3.1.2.6 thread_helper()

```
def pypalex.__main__.thread_helper (
              extractor )
```

Supports multiprocess color extraction operations.

Parameters

extractor	The Extractor object for which to run extraction process.

Returns

The Extractor object.

3.2 pypalex.arg_messages Namespace Reference

Archive of messages to display for arguments supplied by user.

Functions

- def bad_source_message ()

  Returns an error message if the sources provided were not images.
- def bad_directory_message ()

  Returns an error message if the directory provided is not a valid directory.
- def no_args_help_message ()

  Returns a help message if no arguments were presented.

3.2.1 Detailed Description

Archive of messages to display for arguments supplied by user.

Author

Al Timofeyev

Date

March 3, 2022

3.2.2 Function Documentation

3.2.2.1 bad_directory_message()

```
def pypalex.arg_messages.bad_directory_message ( )
```

Returns an error message if the directory provided is not a valid directory.

Returns

The "bad directory" message.

3.2.2.2 bad_source_message()

```
def pypalex.arg_messages.bad_source_message ( )
```

Returns an error message if the sources provided were not images.

Returns

The "bad sources" message.

3.2.2.3 no_args_help_message()

```
def pypalex.arg_messages.no_args_help_message ( )
```

Returns a help message if no arguments were presented.

Returns

The "no arguments" help message.

3.3 pypalex.conversion_utils Namespace Reference

Utilities for converting between RGB, HSL, HEX.

Functions

- def rgb_to_hsl (rgb_array)
  Convert rgb array [r, g, b] to hsl array [h, s, l].
- def hsl_to_rgb (hsl_array)
  Convert hsl array [h, s, l] to rgb array [r, g, b].
- def rgb_to_hex (rgb_array)
  Convert rgb array [r, g, b] to hex string 'ffffff'.

3.3.1 Detailed Description

Utilities for converting between RGB, HSL, HEX.

Author

Al Timofeyev

Date

February 2, 2022

3.3.2 Function Documentation

3.3.2.1 hsl_to_rgb()

```
def pypalex.conversion_utils.hsl_to_rgb (
              hsl_array )
```

Convert hsl array [h, s, l] to rgb array [r, g, b].

HSL where h is in the set [0, 359] and s, l are in the set [0.0, 100.0]. RGB where r, g, b in the set [0, 255]. Formula adapted from https://www.rapidtables.com/convert/color/hsl-to-rgb.html

Parameters

hsl_array	HSL array [h, s, l].

Returns

RGB array [r, g, b].

3.3.2.2 rgb_to_hex()

```
def pypalex.conversion_utils.rgb_to_hex (
              rgb_array )
```

Convert rgb array [r, g, b] to hex string 'ffffff'.

RGB where r, g, b are in the set [0, 255]. Hex string in set ["000000", "ffffff"].

Parameters

rgb_array	RGB array [r, g, b].

Returns

Hex string 'ffffff'

3.3.2.3 rgb_to_hsl()

```
def pypalex.conversion_utils.rgb_to_hsl (
              rgb_array )
```

Convert rgb array [r, g, b] to hsl array [h, s, l].

RGB where r, g, b are in the set [0, 255]. HSL where h in the set [0, 359] and s, l in the set [0.0, 100.0]. Formula adapted from https://www.rapidtables.com/convert/color/rgb-to-hsl.html

Parameters

rgb_array	RGB array [r, g, b].

Returns

HSL array [h, s, l].

3.4 pypalex.extraction_utils Namespace Reference

Utilities for extracting colors from the image.

Functions

- def extract_ratios (hsl_img_array)

  Extracts the ratios of hues per pixel.
- def construct_base_color_dictionary (hsl_img_array)

  Constructs dictionary of base colors from array of hsl pixel values.
- def check_missing_colors (base_color_dict)

  Checks for any missing colors in the base color dictionary and borrows them from the surrounding colors.
- def borrow_for_color_red (base_color_dict, from_left, from_right)

  Borrows colors for the base color red.
- def borrow_for_color_yellow (base_color_dict, from_left, from_right)

  Borrows colors for the base color yellow.
- def borrow_for_color_green (base_color_dict, from_left, from_right)

  Borrows colors for the base color green.
- def borrow_for_color_cyan (base_color_dict, from_left, from_right)

  Borrows colors for the base color cyan.
- def borrow_for_color_blue (base_color_dict, from_left, from_right)

  Borrows colors for the base color blue.
- def borrow_for_color_magenta (base_color_dict, from_left, from_right)

  Borrows colors for the base color magenta.
- def extract_dominant_colors (base_color_dict)

  Extracts dominant light, normal, dark colors from each of the base colors.
- def extract_colors (hsl_base_color_array)

  Extracts the dominant light, normal, dark colors from the color array.
- def sort_by_light_value (hsl_base_color_array)

  Sorts the colors array by the lightness value and returns three separate color arrays.
- def extract_dominant_color (hsl_colors)

  Extracts the dominant color from the hsl color array.
- def get_dom_hue_colors (hsl_colors)

  Construct list/array of a base color with the dominant hue.
- def get_dom_sat_colors (hsl_colors)

  Construct list/array of a base color with the dominant saturation.
- def get_dom_lit_colors (hsl_colors)

  Construct list/array of a base color with the dominant lightness.

- def check_colors (light_color, norm_color, dark_color)

  Checks to make sure all the color types have been properly set by <extract_colors()> function.
- def check_sat_and_light (hsl_color)

  Normalize saturation and lightness so that saturation isn't completely 0% and that lightness isn't 0% or 100%.
- def generate_remaining_colors (dom_color_dict, ratios)

  Generate the remaining black and white, and background and foreground colors.
- def get_color_extremes (ratios)

  Determines the most and least dominant color in the image.
- def generate_black_and_white (hsl_color)

  Generate a black and white color using the hsl_color.
- def generate_background_and_foreground (most_dom_hsl_color, least_dom_hsl_color)

  Generates the background and foreground colors based on the most and least dominant colors.

3.4.1 Detailed Description

Utilities for extracting colors from the image.

Author

Al Timofeyev

Date

February 10, 2022

3.4.2 Function Documentation

3.4.2.1 borrow_for_color_blue()

```
def pypalex.extraction_utils.borrow_for_color_blue (
              base_color_dict,
              from_left,
              from_right )
```

Borrows colors for the base color blue.

Parameters

base_color_dict	Dictionary with arrays of all the base colors.
from_left	Boolean flag for recursive calls, if we came from the left.
from_right	Boolean flag for recursive calls, if we came from the right.

Returns

A list/array sample of a potential color substitute.

3.4.2.2 borrow_for_color_cyan()

```
def pypalex.extraction_utils.borrow_for_color_cyan (
              base_color_dict,
              from_left,
              from_right )
```

Borrows colors for the base color cyan.

Parameters

base_color_dict	Dictionary with arrays of all the base colors.
from_left	Boolean flag for recursive calls, if we came from the left.
from_right	Boolean flag for recursive calls, if we came from the right.

Returns

A list/array sample of a potential color substitute.

3.4.2.3 borrow_for_color_green()

```
def pypalex.extraction_utils.borrow_for_color_green (
              base_color_dict,
              from_left,
              from_right )
```

Borrows colors for the base color green.

Parameters

base_color_dict	Dictionary with arrays of all the base colors.
from_left	Boolean flag for recursive calls, if we came from the left.
from_right	Boolean flag for recursive calls, if we came from the right.

Returns

A list/array sample of a potential color substitute.

3.4.2.4 borrow_for_color_magenta()

```
def pypalex.extraction_utils.borrow_for_color_magenta (
              base_color_dict,
              from_left,
              from_right )
```

Borrows colors for the base color magenta.

Parameters

base_color_dict	Dictionary with arrays of all the base colors.
from_left	Boolean flag for recursive calls, if we came from the left.
from_right	Boolean flag for recursive calls, if we came from the right.

Returns

A list/array sample of a potential color substitute.

3.4.2.5 borrow_for_color_red()

```
def pypalex.extraction_utils.borrow_for_color_red (
              base_color_dict,
              from_left,
              from_right )
```

Borrows colors for the base color red.

Parameters

base_color_dict	Dictionary with arrays of all the base colors.
from_left	Boolean flag for recursive calls, if we came from the left.
from_right	Boolean flag for recursive calls, if we came from the right.

Returns

A list/array sample of a potential color substitute.

3.4.2.6 borrow_for_color_yellow()

```
def pypalex.extraction_utils.borrow_for_color_yellow (
              base_color_dict,
              from_left,
              from_right )
```

Borrows colors for the base color yellow.

Parameters

base_color_dict	Dictionary with arrays of all the base colors.
from_left	Boolean flag for recursive calls, if we came from the left.
from_right	Boolean flag for recursive calls, if we came from the right.

Returns

A list/array sample of a potential color substitute.

3.4.2.7 check_colors()

```
def pypalex.extraction_utils.check_colors (
              light_color,
              norm_color,
              dark_color )
```

Checks to make sure all the color types have been properly set by <extract_colors()> function.

Parameters

light_color	Light hsl color.
norm_color	Normal hsl color.
dark_color	Dark hsl color.

3.4.2.8 check_missing_colors()

```
def pypalex.extraction_utils.check_missing_colors (
              base_color_dict )
```

Checks for any missing colors in the base color dictionary and borrows them from the surrounding colors.

Parameters

base_color_dict	Dictionary with arrays of all the base colors.

3.4.2.9 check_sat_and_light()

```
def pypalex.extraction_utils.check_sat_and_light (
              hsl_color )
```

Normalize saturation and lightness so that saturation isn't completely 0% and that lightness isn't 0% or 100%.

Parameters

hsl_color	An hsl color in list/array format [h, s, l].

3.4.2.10 construct_base_color_dictionary()

```
def pypalex.extraction_utils.construct_base_color_dictionary (
              hsl_img_array )
```

Constructs dictionary of base colors from array of hsl pixel values.

Parameters

hsl_img_array	2D Array of pixels in hsl array format.

Returns

Dictionary of base colors.

3.4.2.11 extract_colors()

```
def pypalex.extraction_utils.extract_colors (
              hsl_base_color_array )
```

Extracts the dominant light, normal, dark colors from the color array.

Parameters

hsl_base_color_array	2D Array of hsl colors from one of the base colors.

Returns

List/Array of dominant light, normal, dark colors in hsl format.

3.4.2.12 extract_dominant_color()

```
def pypalex.extraction_utils.extract_dominant_color (
              hsl_colors )
```

Extracts the dominant color from the hsl color array.

Parameters

hsl_colors	2D Array of hsl colors from one of the base colors [Red, Green, Blue, etc.].

Returns

A dominant color list/array in hsl format [h, s, l].

3.4.2.13 extract_dominant_colors()

```
def pypalex.extraction_utils.extract_dominant_colors (
              base_color_dict )
```

Extracts dominant light, normal, dark colors from each of the base colors.

Parameters

base_color_dict	Dictionary with arrays of all the base colors.

Returns

Dictionary of light, normal, dark colors for each of the base colors.

3.4.2.14 extract_ratios()

```
def pypalex.extraction_utils.extract_ratios (
              hsl_img_array )
```

Extracts the ratios of hues per pixel.

Parameters

hsl_img_array	2D Array of pixels in hsl array format.

Returns

Dictionary of hue ratios (percentage) in set [0.000, 100.000]

3.4.2.15 generate_background_and_foreground()

```
def pypalex.extraction_utils.generate_background_and_foreground (
              most_dom_hsl_color,
              least_dom_hsl_color )
```

Generates the background and foreground colors based on the most and least dominant colors.

Parameters

most_dom_hsl_color	The hsl color array from which to generate background.
least_dom_hsl_color	The hsl color array from which to generate foreground.

Returns

List/array of background and foreground hsl colors.

3.4.2.16 generate_black_and_white()

```
def pypalex.extraction_utils.generate_black_and_white (
              hsl_color )
```

Generate a black and white color using the hsl_color.

Parameters

hsl_color	The hsl color array from which to generate black and white.

Returns

List/array of light, normal, dark black and white hsl colors.

3.4.2.17 generate_remaining_colors()

```
def pypalex.extraction_utils.generate_remaining_colors (
              dom_color_dict,
              ratios )
```

Generate the remaining black and white, and background and foreground colors.

Parameters

dom_color_dict	Dictionary of dominant light, normal, dark base colors.
ratios	Dictionary of ratios of the base colors in image.

3.4.2.18 get_color_extremes()

```
def pypalex.extraction_utils.get_color_extremes (
              ratios )
```

Determines the most and least dominant color in the image.

Parameters

ratios	Dictionary of ratios of the base colors in image.

Returns

List/array of most and least dominant color as strings.

3.4.2.19 get_dom_hue_colors()

```
def pypalex.extraction_utils.get_dom_hue_colors (
              hsl_colors )
```

Construct list/array of a base color with the dominant hue.

Example: From the hsl_colors array, there could be 10 hues that appear 4 times each, while the rest of the hues appear only once or twice. The hsl_colors with these 10 hues will be extracted and returned because they appear the most and are therefore the dominant hues.

Parameters

hsl_colors	2D Array of hsl colors from one of the base colors [Red, Green, Blue, etc.].

Returns

List/array of all hsl colors that had the dominant number of hue values.

3.4.2.20 get_dom_lit_colors()

```
def pypalex.extraction_utils.get_dom_lit_colors (
              hsl_colors )
```

Construct list/array of a base color with the dominant lightness.

Example: From the hsl_colors array, there could be 2 lightness values that appear 8 times each, while the rest of the lightness values appear only once or twice. The hsl_colors with these 2 lightness values will be extracted and returned because they appear the most and are therefore the dominant lightness values.

Parameters

hsl_colors	2D Array of hsl colors from one of the base colors [Red, Green, Blue, etc.].

Returns

List/array of all hsl colors that had the dominant number of lightness values.

3.4.2.21 get_dom_sat_colors()

```
def pypalex.extraction_utils.get_dom_sat_colors (
              hsl_colors )
```

Construct list/array of a base color with the dominant saturation.

Example: From the hsl_colors array, there could be 5 saturation values that appear 12 times each, while the rest of the saturation values appear only once or twice. The hsl_colors with these 5 saturation values will be extracted and returned because they appear the most and are therefore the dominant saturation values.

Parameters

hsl_colors	2D Array of hsl colors from one of the base colors [Red, Green, Blue, etc.].

Returns

List/array of all hsl colors that had the dominant number of saturation values.

3.4.2.22 sort_by_light_value()

```
def pypalex.extraction_utils.sort_by_light_value (
              hsl_base_color_array )
```

Sorts the colors array by the lightness value and returns three separate color arrays.

Parameters

hsl_base_color_array	2D Array of hsl colors from one of the base colors.

Returns

List/Array of light, normal, and dark colors from the array of hsl colors.

3.5 pypalex.Extractor Namespace Reference

Extraction utility class for extracting colors from the image.

Classes

- class Extractor

  Extracts colors using ONLY the colors in the image.

3.5.1 Detailed Description

Extraction utility class for extracting colors from the image.

Author

Al Timofeyev

Date

February 10, 2022

3.6 pypalex.image_utils Namespace Reference

Utilities for processing image and file handling.

Functions

- def process_image (image)
  Processes PIL Image object.
- def rescale_image (img)
  Rescales image to a smaller sampling size.
- def thread_helper (flat_img_array)
  Helper function for multiprocessing conversion operations.
- def save_palette_to_file (color_palette, output_file)
  Saves color palette to json file.

3.6.1 Detailed Description

Utilities for processing image and file handling.

Author

Al Timofeyev

Date

February 27, 2022

3.6.2 Function Documentation

3.6.2.1 process_image()

```
def pypalex.image_utils.process_image (
            image )
```

Processes PIL Image object.

Multiprocessing example from: https://stackoverflow.com/a/45555516

Parameters

image	PIL Image object.

Returns

List of full hsl arrays (pixels) of image.

3.6.2.2 rescale_image()

```
def pypalex.image_utils.rescale_image (
              img )
```

Rescales image to a smaller sampling size.

Parameters

img	The PIL.Image object.

Returns

Tuple of the new width and height of image.

3.6.2.3 save_palette_to_file()

```
def pypalex.image_utils.save_palette_to_file (
              color_palette,
              output_file )
```

Saves color palette to json file.

If a file with the same name already exists, it is overwritten.

Parameters

color_palette	Dictionary of light, normal, and dark color palettes.
output_file	The output path/directory with filename at the end.

3.6.2.4 thread_helper()

```
def pypalex.image_utils.thread_helper (
              flat_img_array )
```

Helper function for multiprocessing conversion operations.

Helps convert from [r, g, b] to [h, s, l].

Parameters

flat_img_array	A flattened rgb portion of the original image array.

Returns

A numpy array of converted hsl values.

Chapter 4

Class Documentation

4.1 pypalex.Extractor.Extractor Class Reference

Extracts colors using ONLY the colors in the image.

Public Member Functions

- def __init__ (self, full_hsl_img_array, output_path, pastel=False, pastel_light=False, pastel_normal=False, pastel_dark=False)

  Extractor Constructor.
- def run (self)

  Performs extraction of colors.
- def construct_palette_dictionary (self)

  Constructs color palette dictionary.
- def check_pastel_conversion (self)

  Checks to see if any of the palettes should be converted to pastel.
- def convert_pastel (self, hsl_color)

  Converts/normalizes hsl color to pastel.
- def convert_pastel_light (self)

  Converts light palette to pastel.
- def convert_pastel_normal (self)

  Converts normal palette to pastel.
- def convert_pastel_dark (self)

  Converts dark palette to pastel.

Public Attributes

- **output_path**

  Output path and filename of where to store color palette.
- **full_hsl_img_array**

  A 2D numpy array of all the pixels from image, in hsl format.
- **pastel**
- **pastel_light**
- **pastel_normal**
- **pastel_dark**
- **ratio_dict**
- **base_color_dict**
- **dom_color_dict**

4.1.1 Detailed Description

Extracts colors using ONLY the colors in the image.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 __init__()

```
def pypalex.Extractor.Extractor.__init__ (
              self,
              full_hsl_img_array,
              output_path,
              pastel = False,
              pastel_light = False,
              pastel_normal = False,
              pastel_dark = False )
```

Extractor Constructor.

Parameters

self	The object pointer.
full_hsl_img_array	A 2D numpy array of all the pixels from image, in hsl format.
output_path	Output path and filename of where to store color palette.
pastel	Flag to convert all extracted color palettes to pastel.
pastel_light	Flag to convert light palette to pastel.
pastel_normal	Flag to convert normal palette to pastel.
pastel_dark	Flag to convert dark palette to pastel.

4.1.3 Member Function Documentation

4.1.3.1 check_pastel_conversion()

```
def pypalex.Extractor.Extractor.check_pastel_conversion (
              self )
```

Checks to see if any of the palettes should be converted to pastel.

Parameters

self	The object pointer.

4.1.3.2 construct_palette_dictionary()

```
def pypalex.Extractor.Extractor.construct_palette_dictionary (
              self )
```

Constructs color palette dictionary.

Parameters

self	The object pointer.

4.1.3.3 convert_pastel()

```
def pypalex.Extractor.Extractor.convert_pastel (
              self,
              hsl_color )
```

Converts/normalizes hsl color to pastel.

For values x in range [a, b], values x can be converted to the new range [y, z] with the following equation: new_x = (z-y) $*$ ((x-min_x) / (max_x-min_x)) + y where a = min_x and b = max_x from the old range.

Parameters

self	The object pointer.
hsl_color	HSL color to be converted to pastel.

4.1.3.4 convert_pastel_dark()

```
def pypalex.Extractor.Extractor.convert_pastel_dark (
              self )
```

Converts dark palette to pastel.

Parameters

self	The object pointer.

4.1.3.5 convert_pastel_light()

```
def pypalex.Extractor.Extractor.convert_pastel_light (
              self )
```

Converts light palette to pastel.

Parameters

self	The object pointer.

4.1.3.6 convert_pastel_normal()

```
def pypalex.Extractor.Extractor.convert_pastel_normal (
              self )
```

Converts normal palette to pastel.

Parameters

self	The object pointer.

4.1.3.7 run()

```
def pypalex.Extractor.Extractor.run (
              self )
```

Performs extraction of colors.

Parameters

self	The object pointer.

The documentation for this class was generated from the following file:

- Extractor.py

Index